// 矩阵快速幂

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

const int MOD=10000;

struct mat

{

ll a[2][2];

};

mat mat\_mul(mat x,mat y)

{

mat res;

memset(res.a,0,sizeof(res.a));

for(int i=0;i<2;i++)

for(int j=0;j<2;j++)

for(int k=0;k<2;k++)

res.a[i][j]=(res.a[i][j]+x.a[i][k]\*y.a[k][j])%MOD;

return res;

}

void mat\_pow(int n)

{

mat c,res;

c.a[0][0]=c.a[0][1]=c.a[1][0]=1;

c.a[1][1]=0;

memset(res.a,0,sizeof(res.a));

for(int i=0;i<2;i++) res.a[i][i]=1;

while(n)

{

if(n&1) res=mat\_mul(res,c);

c=mat\_mul(c,c);

n=n>>1;

}

printf("%I64d\n",res.a[0][1]);

}

int main()

{

int n;

while(~scanf("%d",&n)&&n!=-1)

mat\_pow(n);

}